

# Benefits to the Simulation Training Community of a New ANSI Standard for the Exchange of Aero Simulation Models

Bruce Hildreth  
SAIC  
Lexington Park, MD  
[Bruce.Hildreth@SAIC.com](mailto:Bruce.Hildreth@SAIC.com)

E. Bruce Jackson  
NASA Langley Research Center  
Hampton, VA  
[Bruce.Jackson@NASA.gov](mailto:Bruce.Jackson@NASA.gov)

## ABSTRACT

The American Institute of Aeronautics Astronautics (AIAA) Modeling and Simulation Technical Committee is in final preparation of a new standard for the exchange of flight dynamics models. The standard will become an ANSI standard and is under consideration for submission to ISO for acceptance by the international community.

The standard has some aspects that should provide benefits to the simulation training community. Use of the new standard by the training simulation community will reduce development, maintenance and technical refresh investment on each device. Furthermore, it will significantly lower the cost of performing model updates to improve fidelity or expand the envelope of the training device. Higher flight fidelity should result in better transfer of training, a direct benefit to the pilots under instruction. Costs of adopting the standard are minimal and should be paid back within the cost of the first use for that training device.

The standard achieves these advantages by making it easier to update the aerodynamic model. It provides a standard format for the model in a custom eXtensible Markup Language (XML) grammar, the Dynamic Aerospace Vehicle Exchange Markup Language (DAVE-ML). It employs an existing XML grammar, MathML, to describe the aerodynamic model in an input data file, eliminating the requirement for actual software compilation. The major components of the aero model become simply an input data file, and updates are simply new XML input files. It includes naming and axis system conventions to further simplify the exchange of information.

## ABOUT THE AUTHORS

**Bruce L. Hildreth** is a SAIC VP, Technical Fellow, and Senior Program Manager. He holds an M.S. in Aerospace & Mechanical Sciences from Princeton University and a B.S. in Aerospace Engineering from Virginia Tech. His career as an aerospace engineer has focused on the flight testing, flight dynamics, controls, and simulation of aircraft, uninhabited vehicles, missiles, and aviation subsystems. Mr. Hildreth is a past chairman and emeritus member of the AIAA Modeling and Simulation Technical Committee. He leads the standards working group of that committee. He has 19 technical papers on flight testing and simulation and is a senior member of the AIAA.

**E. Bruce Jackson** is a Senior Research Engineer at NASA Langley Research Center. He has an M.S. in Aeronautics and Astronautics from Stanford University and a B.S. in Aeronautical Engineering from NC State University. He has worked in flight simulation since 1982, initially at NAVAIR's Manned Flight Simulator and is currently involved in flight control design and flying qualities research for manned spacecraft. Mr. Jackson has architected three flight simulation frameworks, including the initial versions of CASTLE, LaRCsim and the original EOM for the open-source FlightGear project. Mr. Jackson is an Associate Fellow of the AIAA and a 2009 winner of the NASA Exceptional Service Medal for work in aircraft and spacecraft flying qualities and flight simulation technology.

# Benefits to the Simulation Training Community of a New ANSI Standard for the Exchange of Aero Simulation Models

Bruce Hildreth  
SAIC  
Lexington Park, MD  
[Bruce.Hildreth@SAIC.com](mailto:Bruce.Hildreth@SAIC.com)

E. Bruce Jackson  
NASA Langley Research Center  
Hampton, VA  
[Bruce.Jackson@NASA.gov](mailto:Bruce.Jackson@NASA.gov)

## INTRODUCTION

There is a new standard that should provide significant benefits to the training simulation community. It is in final preparation by the American Institute of Aeronautics and Astronautics, the largest aerospace professional society in the United States. It is being prepared as an American National Standards Institute (ANSI) standard by the Modeling and Simulation Technical Committee (MSTC) of the AIAA. After ANSI acceptance it is planned to submit it as an International Organization for Standardization (ISO) standard.

The purpose of this standard is to provide a method to encode and exchange simulation models. This method, called DAVE-ML, is accomplished through a custom XML grammar. DAVE-ML provides the format and structure to exchange models, primarily aerodynamic equations and constants; and function tables in general.

Furthermore it defines a standard set of axis systems and variable names to help the user transfer information from one simulation to another. The axis systems and variable names provide the definition of the information exchanged. They are essentially the “common language” to use when exchanging models. Figure 1 illustrates how three sets of information describing the model (common physics, common language, common data) are required for the exchange of a simulation model. It is not enough to simply exchange data. The exporter and the importer of the model must also clearly communicate the information that is exchanged (a Common Language) and also understand enough of each other’s physics (Common Physics). A simple example of “Common Physics” is: does the importer understand that the exporter’s simulation is based on a flat earth formulation of the equations of motion?

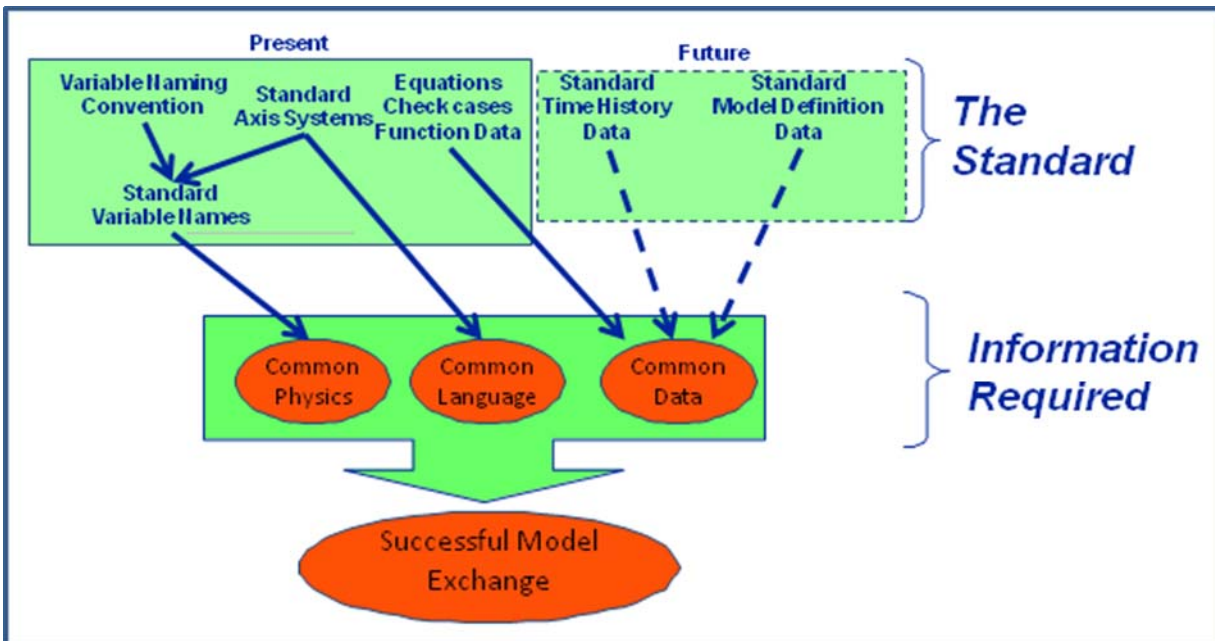


Figure 1. The three critical elements for a model standard

The standard does not require use of the standard axes (part of Common Physics) or variable names (part of Common Language); the user can define their own. However use of the standard names and axes frees the user from having to define and clearly document it themselves, which is a large task.

## BENEFITS TO THE SIMULATION COMMUNITY

The core motivation behind developing a standard was to facilitate collaboration between simulators and simulations. The simulation community needs to be more efficient and effective. The advantage of the standard can best be illustrated by an analyses of actual simulations related to a specific weapon system (Jackson and Hildreth, 2002).

There are many simulations of this specific aircraft:

- Prime manufacturer developmental simulations (manned and unmanned)
- DoD RDT&E simulations (manned and unmanned)
- NASA RDT&E simulations (manned and unmanned)
- Customer country simulations (manned and unmanned)
- Part task crew training simulations (manned)
- Full fidelity crew training simulations (manned)
- Crew training mission capable simulations (manned)

Usually there are many different simulation architectures, languages, validation processes, etc., for simulations of the same aircraft. Most of these simulations have different goals and objectives. However at various levels, they have some commonality. In general, for example, they all require an aerodynamic model, a controls model, and an engine model. These models may be of different fidelity but they all must have some of the same physics and data.

Presently when a team working one of the simulations finds an error or makes an improvement, it is very difficult to share this with all the other simulations, actually it is nearly impossible. The barriers to cooperation are the fact that each simulation has its own architecture.

Since it is virtually impossible to have governments and industries standardize all architectures (also probably a bad idea for reasons not within the scope of this paper), the MSTC arrived instead at the solution of a common model **exchange format** (Figure 2). If each

of the simulation teams adopts this exchange format then they simply have to write simple import and export applications once to exchange simulation models between all other simulation teams. This applies to any aircraft model and type. It is likely each of these teams creates export and import routines to exchange with any other simulation team anyway. The standard just creates one format for all to use. This saves significant effort. Additionally, this standard will facilitate collaboration, which is the true goal and provides the greatest benefit.

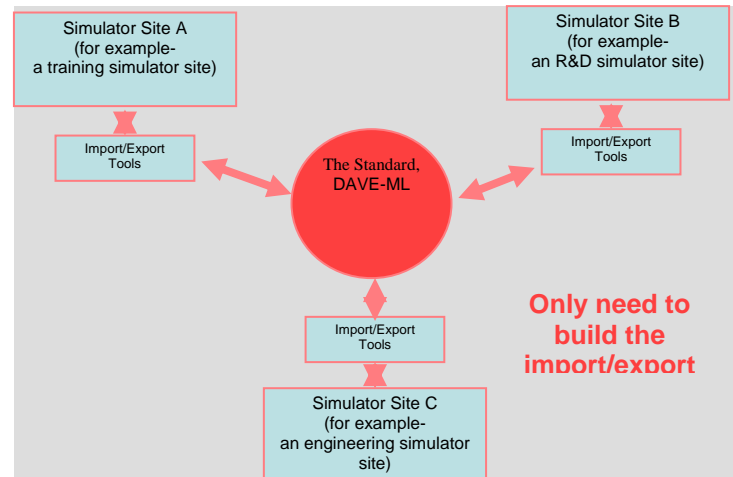


Figure 2. The exchange standard can be used by any simulation architecture

There are four major areas in the simulation training community that are most benefitted by the standard:

- development of the simulation
- software maintenance or updates
- tech refresh
- improved fidelity resulting in improved transfer of training

## Reducing the cost and time to develop the simulation

The training simulator typically is started well before the prime system development has been completed. Typically the training simulation aerodynamic databases use some data from the prime system development and develop some of their own. They make changes to the prime system data when they find there is an error or when it does not fully satisfy their training system requirements. The training system developers have to write unique software tools to convert the prime system data (aerodynamic functions, for example) to their architecture.

Because it is so difficult to translate from the prime system to the training simulator (and vice versa) this activity usually only happens a few times (maybe once?) during the prime system development. Consequently the training simulation data and the prime system data diverge.

Using the standard makes transferring information extremely easy, allows reuse of the data, and encourages collaboration. The differences between the two sets of data will be significantly less; both models will be better with the improved collaboration. The resultant trainer model will also be closer to the final aircraft configuration.

### Improved software maintenance

Proper use of the standard also reduces the cost of software maintenance. Typically maintenance is done by a different team than the one that developed the original simulation. Often the maintenance team has to reverse-engineer how to edit or update models and function tables. They have to reverse-engineer the exact definition of simulation variables due to unclear or incomplete documentation. The implementation of the dynamic models may not be completely understood due to axis systems irregularities or confusion over units.

The standard will not eliminate all of these problems but will certainly alleviate most of them in that the use of standard variable names and axis systems simplifies the documentation process and provides a framework which the software maintenance person can work within. The standard includes very clear and complete definitions of these variables including units and sign conventions. How many mistakes have occurred due to software not being in the right units? Furthermore, software maintenance occurs both because deficiencies that have been found in the trainer and modifications have been made to the prime system. Again, the use of the standard would dramatically simplify incorporating changes from prime system simulations or any other source such as government simulation facilities. The configuration of the training device would more closely match that of the prime system in less cost and time.

However, following a convention for indicating states and state derivatives and controls also has huge potential benefits. Mathematically, all simulation variables are calculated from states and controls. In the vast majority of simulations, there is no systematic method to identify these critical variables. In any software modification it is absolutely critical for the software engineers to identify states and state

derivatives. Yet to date, no systematic method exists to do so. The standard naming convention solves this problem by clearly specifying these by a prefix on the variable name.

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

#### (1) Common mathematical definition of states and state derivatives

Take as an example the requirement to modify a simulator to allow the simulation of an of actuator malfunction. To do this requires adding some fidelity improvement to the actuator in order to make it realistically reproduce the malfunction. Assuming the actuator dynamics must be changed to meet the fidelity requirement, how can the standard be used to lower the cost of this modification?

- 1) Cost reduction through reuse of an existing actuator model in a similar simulation. Often in this kind of situation, there are engineering simulations that have already studied the problem, or have an actuator model that is suitable for use. The standard reduces the barriers to reuse of this software. This is due to the fact that the actuator states are clearly defined and easy to find in both simulations.
- 2) Any non-linear functions can be dropped in from the engineering simulation without changing the function tables.
- 3) The sign convention used in variable names reduces the possibility of making errors in the modified actuator due to mismatching units.

### Technology Refresh

There are many aspects of technology refresh such as updating the visuals, motion system, cockpit, etc. but the standard applies primarily to updating the host computers. Updating the host computers often requires recoding /conversion from an obsolete language such as Fortran to a newer language such as C/C++. In this conversion the models and their function tables must be converted also. Use of the standard virtually eliminates the need for recoding the function tables and aerodynamic routines. There are tools available that do this for the user. The model is simply reloaded, not recoded.

### Improve Model Fidelity/ Transfer of Training

The above are primarily cost and schedule benefits. Perhaps the biggest technical benefit of the standard is improved collaboration between all the different types of simulations results in an improved math model on the trainer. For example, in an actual large aircraft program, one facility was concentrating on high angle of attack and developed an improved a high angle of attack database. Another team found an error in the flaps down approach-to-land database. Another simulation developed and improved the engine model and yet another added stores. In that program there was no systematic way to update the training simulations and it is probable the training devices still don't have most of the updates. With the barriers to data transfers that exist now it is rare that these improvements get widely distributed.

The standard significantly removes or eliminates these barriers and would allow all these improvements to be incorporated into the training simulation at little cost. Therefore the simulation will now have a wider flight envelope, improved utility and better fidelity.

### **General Benefits**

The expected benefits to any user (not just to training simulations) from adoption of the standard are stated below.

- Can easily share original aerodynamics models and all function tables or updates to each between research development, flight test, and training simulations of all architectures
- No requirement to rewrite or host any function tables and most any part of the aero model
- Resulting standard file is both human- and computer-readable
- The model is simulation architecture and code agnostic
- The standard avoids proprietary languages and proprietary standards or practices
- The standard is expandable to accommodate new components and features
- The standard is maintainable, it can change as the simulation user's requirements change
- Functions stored in the standard DAVE-ML format have improved documentation and provenance features

### **Cost Benefit**

Those familiar with the translation of models from one simulation architecture to another easily appreciate the

potential for savings created by this standard. The authors of this paper earlier estimated (Jackson & Hildreth, 2002) an annual potential savings of \$6.8M in a case study of simulators for one type aircraft. 30 of the 59 simulations are pilot training simulations, the rest are RDT&E or factory simulations. Jackson and Hildreth (2002) also state that the biggest potential benefit is from increased collaboration between the people working on the different simulations. It is difficult to qualify the true gains in time and money, but the authors take the position that the leverage on improved collaboration is huge. A small gain in collaboration results in a great benefit to the training simulation community.

### **Example Application in the Next Generation Threat system (NGTS)**

One of first training system applications of the standard is in the implantation of the aerodynamic model of NGTS. NGTS is a constructive simulation of many friendly, neutral, and threat aircraft (Hildreth, Linse & Dicola, 2008). It is a joint US Navy, Marine Corps, and Air Force system. The models have realistic performance but have simplified equivalent system dynamics. Realistic aerodynamic based vehicle moments and dynamics are not included. The models include several physics based geometry, size, and mass parameters and several one and two dimensional aerodynamic tables. The equivalent system models require additional parameters. All of the model parameter and tables are implemented in DAVE-ML using Janus application software (Australian Government, DSTO, Janus..., 2009).

The advantages seen by NGTS from using DAVE-ML include:

- the heavily documented data references that have been incorporated into the parameter and function table structure of DAVE-ML provide for better model documentation.
- NGTS attempts to use only models previously validated. As additional threat, friendly, and neutral aircraft models are added, they may come from other sources for which the model structure would not be expected to be the same. Since DAVE-ML allows the model to be built up in MathML, these new model structures and function tables can be accommodated without any recoding inside NGTS, only the XML and MathML data are changed.

In general, the model structure flexibility allows updating the models with only data changes, no code

changes are required. This makes it extremely easy to add new models to the environment since they often have different model structures.

### **Example of an update to a full fidelity flight simulator**

As an example of the advantages of using the standard as implemented by DAVE-ML, consider updating the aero model of a full flight mission training simulator. Specifically, consider updating the aerodynamic database. Aerodynamic database updates are infrequently performed on typical fleet pilot trainers. This is due to the significant cost of such updates. As a result the pilots under instruction are actually flying simulators with reduced fidelity because keeping them updated with the latest data simply costs too much and takes too long. The standard would tremendously reduce the cost and time to make simple updates such as modifying, adding, or deleting aerodynamic function tables from the models. This is because in general no code changes would be required. The XML specified model would simply be edited and the new data tables are updated with new or modified data. In fact, if all the simulators of that type of aircraft used for crew training would accept data from the standard, all would logically be updated with the same database.

More significantly, adoption of the standard would simplify validation of the models. All of the models would have one set of validation data that applied to all the trainers. This would realize a tremendous saving in cost, and would give the pilots under instruction access to the best possible simulator, not one that is out of date.

## **THE NEW ANSI STANDARD**

At the time of this writing, the AIAA is accepting public comment on this proposed standard, Flight Dynamic Model Exchange Standard, BSR/AIAA S-119 (NASA, Flight Dynamic..., 2009). The benefits of this standard were described previously in this paper; the rationale for and provisions of the new standard are described below.

### **Paucity of existing modeling standards**

Despite the growth of simulation standard organizations, such as the Simulation Interoperability Standards Organization (SISO), and the interest in having simulations networked together for mission training and war games, very little attention has been focused on the exchange of flight dynamic models.

The ANSI and the AIAA previously adopted a Recommended Practice (1992) that addresses common nomenclature for axis systems and goes so far as to specify Greek symbols for use in documentation of flight dynamic models; it also describes a best practice for use of quaternions in digital equations of motion. However, it (unfortunately) does not specify ASCII or UNICODE variable names, units of measure, or any implementation details for flight dynamic models (aside from the quaternion hints). The result has been fifteen more years without a universally agreed, clearly defined method for describing the mathematics of flight dynamics in software. This deficiency was highlighted by Hildreth (1994) in one of the earliest papers formulating the concept of a model exchange standard.

### **Philosophy**

The standard is intended to be an exchange standard for sharing flight dynamic models of aerospace vehicles. It defines a reference set of axis systems based on the ANSI/AIAA Recommended Practice (1992) and outlines a method for constructing unambiguous variable names (as well as providing a default set of variable names). The variable names also indicate units of measure in which the value is expressed, for rigor. Further, using a special-purpose grammar based on the W3C eXtensible Markup Language, (W3C, XML, 2009) and reutilizing another grammar for embedded mathematical formulae, MathML, (W3C, MATHML, 2009) a completely self-contained and self-documenting human- and machine-readable encoding scheme is specified. This has been named Digital Aerospace Vehicle Exchange Markup Language (DAVE-ML).

The encoded models represent the vehicle-specific portion of an aerospace vehicle's flight characteristics. It is important to note that the standard does not include a set of equations of motion (state integrations and physical constraints of motion); these are invariant and do not require encoding. The standard encompasses those portions of a model that describe how flight conditions and force/moment effectors affect the forces and moments that maneuver the vehicle.

The contents of a properly structured DAVE-ML model are described below. It is fully defined in Annex B of the standard online (NASA, *Schema...*, 2008).

### **Function Tables**

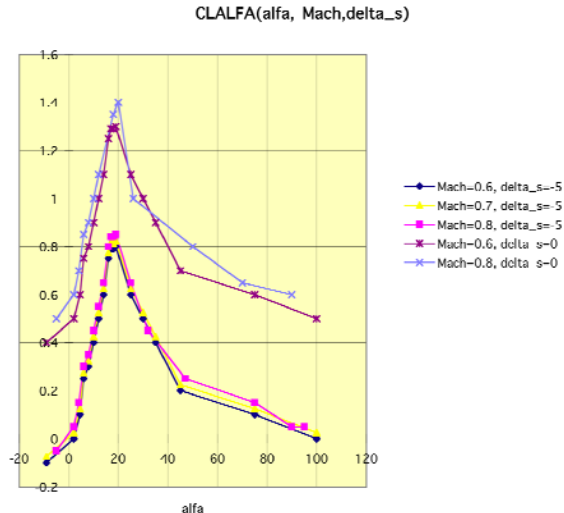
The bulk of most aerospace vehicle flight dynamics models are composed of multi-dimensional tabular data

of aerodynamic coefficients. These tables provide dependent function values tied with specific independent function arguments, and require multidimensional interpolation to find the function value at intermediate points.

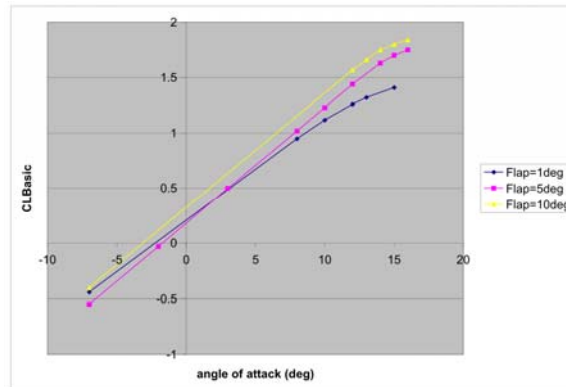
The dependent values normally represent the contribution to a force or moment by some aspect of the vehicle, including basic outer-mold-line shapes and control surface deflections or dynamic damping terms. Tables are also specified for the non-linear functions describing control surface interactions (such as the effect of a canard surface on a wing or tailplane) and reaction-control-system plume impingement on a launching or reentering spacecraft.

It is common practice as well to reuse a single table for several functions. This reuse of tables and the independent table coordinate vectors, also known as breakpoint sets, is supported by DAVE-ML.

The description above describes a ‘gridded’ table in which the function value is given for each possible coordinate set. DAVE-ML also allows specification of ‘ungridded’ tables that provide sets of independent variable values associated with a dependent value scattered throughout the function space. Examples of each type of function are shown in the figures below.



**Figure 3. Example of a “gridded” three-dimensional lift curve as a function of angle of attack, Mach number and control surface deflection.**



**Figure 4. Example of an “ungridded” function with different angle of attack breakpoints for each lift coefficient curve.**

Finally, DAVE-ML functions may also be described mathematically using any arbitrary mathematical expression, including any order polynomial, in place of or in addition to function tables. This makes use of the MathML mathematical markup language elements.

### Mathematical Expressions

Another element of all flight dynamic models are the so-called “build-up equations” that specify how the result of tabular function interpolation or other values are combined to form total coefficients of forces and moments (or in some cases, the actual values of forces and moments). Or the build-up equations might yield new arguments for additional function tables. DAVE-ML supports any arbitrary mix of these elements in mathematical expressions using the MathML mathematical content grammar, a prefix style notation for mathematical operations.

### Provenance

An important feature often neglected in simulation data packages is the history, or provenance, of the model and the various data sources. The proposed standard attempts to address this by including special elements for recording modifications and data sources in the markup grammar. These elements can be used to annotate documents and plots prepared from a model realized in DAVE-ML.

### Axis Systems

A set of the AIAA Recommended Practice (1994) axis systems are re-used and expanded upon in the standard, which assigns two-character encoding to the primary sets of axes (e.g., GE for geocentric earth-fixed). These two-character encodings are then used in variable names to remove ambiguity, where necessary. These encompass most conventional aerospace axis systems.



## Variable names

A set of standard variable names is given for common aerospace measures (e.g. angleOfAttack\_deg, MachNumber); a method is suggested for “expanding the vocabulary,” that is, creating specialized ones for specific vehicles as well. Equally important is a post-fixing system for specifying units of measure for the quantity represented to remove ambiguity.

It is important to note that these variable names are unlikely to be used directly in any heritage physics engine simulation framework; when the capability to use DAVE-ML encoded models in such a legacy framework is added, the mapping of these standard variable names into facility-specific names (such as angleOfAttack\_deg matching AOADEG) will be required.

## Checkcase Data

One of the biggest benefits to adopting the standard is the ability to include checkcase data, which allows for automatic verification of the proper implementation of the model by the importing facility. These data are built up from any number of static input/output vector pairings. Encoded in the DAVE-ML model are specified values for all input parameters and a corresponding expected set of output values, along with a required tolerance for matching. Thus, verification can be performed automatically upon instantiation of the DAVE-ML model at a new facility without the tedious cut-and-paste from text or spreadsheet data sets as is now customary.

## Existing tools to assist with DAVE-ML

At present, the most helpful DAVE-ML compatible software is offered courtesy of the Australian Defence Science Technology Organization (DSTO, Janus..., 2009). It is a C++-based, open-source API library named Janus that reads (and can write) DAVE-ML models at run-time.

An open-source Java-based DAVE-ML parser that can output Matlab® Simulink® block diagrams, DAVEtools, is available from NASA Langley Research Center (NASA, Open-Source..., 2009)

## FUTURE WORK AND IMPROVEMENTS ON THE STANDARD

As discussed in the introduction, the standard is in the final stages of being approved as an American National Standard. The AIAA, which is accredited by ANSI to

develop standards, will also help submit this standard to ISO.

While the current standard addresses static models (such as aerodynamic databases) including verification data (e.g. static shots), work to extend the standard to encode fully dynamic models is desirable. One of the tasks is to encode relatively large time-history verification data into a standard format.

The MSTC has tentatively decided to adopt and tailor an emerging flight test time-history data format. This is the format chosen by the Joint Strike Fighter program. The time-history format would be a subset of this flight test data format. It is implemented in Hierarchical Data Format 5 (HDF 5). HDF is a publically released architecture for storing large amounts of data of all types. Our simulation applications are relatively simple compared to the HDF tools and products available. HDF was started by the National Center for Supercomputing Applications and the University of Illinois at Urbana-Champaign and is now maintained by “The HDF Group” (2009), a non-profit organization. It was not included in the present version of the standard because of the difficulty of getting public release of specific flight test related tools that greatly facilitate the use HDF 5 in our applications. It is hoped that these issues will be shortly overcome and the next iteration of the standard will include “exchange of validation data”.

After this minor hurdle is overcome, a more difficult task looms. How do we standardize on modeling dynamic systems? The proposed standard supports static systems, those with no internal states; a work-around requires external integration of any state derivatives. Control systems and engine models, both dynamic systems, are the most obvious examples of models that need to be exchanged between simulation facilities.

Simulink®, by the Mathworks®, is a leading provider of simulation of diverse dynamic systems. In standardizing the exchange of dynamic system models, the MSTC is considering whether this dominance is the equivalence of a “de facto” standard. Why create something that already exists and has wide acceptance? If DAVE-ML were to include dynamic systems as part of its exchange standard, it would be an XML schema to capture the information contained on Simulink diagrams. Informal discussions on this topic have taken place and will continue.



## SUMMARY

The training simulation community can benefit greatly from adopting the “Flight Dynamics Model Exchange Standard.” The costs of adopting it are very low. If you need to import models from other simulation architectures, the cost of importing them via DAVE-ML are decreasing due to the growing set of tools available. If you are not working at all with other simulation facilities, maybe you should be.

The benefits are many: cost savings at all phases of simulation development and support, improved model fidelity, improved documentation, and most importantly but most difficult to quantify, improved collaboration between simulators and simulation personnel.

## ACKNOWLEDGEMENTS

The authors would like to acknowledge the contributions, encouragement and helpful suggestions from Geoff Brian (Australia's DSTO), Brent York (Indra), Bill Cleveland (NASA Ames), Dennis Linse (originally SAIC, now Vuelo Software Analysis), Bimal Aponso (NASA Ames), Jon Berndt (Jacobs Sverdrup), Mike Silvestro (Draper Labs), J. Dana McMinn (NASA Langley), Peter Grant (UTIAS), Giovanni A. Cignoni (University of Pisa), Daniel M. Newman (formerly Ball Aerospace, now Quantitative Aeronautics), Hilary Keating (Fortburn Pty. Ltd.), Riley Rainey (SDS International), Jeremy Furtek (Delphi Research) and Randy Brumbaugh (Indigo Innovations).

## REFERENCES

American Institute of Aeronautics and Astronautics (1992). *Recommended Practice: Atmospheric and Space Flight Vehicle Coordinate Systems*, ANSI/AIAA R-004-1992.

Australian Department of Defense Science and Technology Organization(DSTO) (2006). *Janus Dynamic Aerospace Vehicle Exchange Markup Language (DAVE-ML) C++ Interpreter*. Retrieved 22 June 2009 from <http://www.dsto.defence.gov.au/research/4675/>

HDF Group. *HDF5 Home Page*. Retrieved on 24 June 2009 from <http://www.hdfgroup.org/HDF5/>.

Hildreth, B. L., Linse, D. J., Dicola, J., (2008). *Pseudo Six Degree of Freedom (DOF) Models for Higher Fidelity Constructive Simulations*. AIAA Paper 2008-6857, AIAA Modeling and Simulation Technology Conference, Honolulu, Hawaii.

Hildreth, Bruce L. (1994). *A Process for the Development of Simulation Standards*. AIAA Paper 94-3430. AIAA Flight Simulation Technology Conference, Baltimore, MD.

Jackson, E. Bruce; and Hildreth, Bruce L. (2002). *Flight Dynamic Model Exchange using XML*. AIAA Paper 2002-4482, AIAA Modeling and Simulation Technology Conference, Monterey, California.

NASA (2002). *Flight Dynamic Model Exchange Using XML*. Retrieved 24 June 2009 from <http://daveml.nasa.gov>.

NASA (2008). *Schema documents associated with DAVE-ML, Version 2.0RC2, 27 October 2008*. From <http://daveml.nasa.gov/DTDs.html>.

NASA. (2009). *NASA Open Source License Agreement for DAVEtools*. Retrieved on 24 June 2009 from <http://dcb.larc.nasa.gov/Products/SW/DAVEtools.html>

World Wide Web Consortium (2006). *W3C Recommendation: Extensible Markup Language (XML) 1.0 Fifth Edition*. Retrieved 22 June 2009 from <http://www.w3.org/TR/xml/>.

World Wide Web Consortium (2001) *W3C Recommendation: Mathematical Markup Language (MathML) Version 2.0 (Second Edition)*. Retrieved 22 June 2009 from <http://www.w3c.org/TR/MathML/>.